

# 一种选择多个单元的重新播种内建自测试方法

马 俊

(安庆师范学院 教育技术系, 安徽 安庆 246011)

**摘 要:**随着集成电路技术的迅速发展,芯片的集成度越来越高,怎样对电路进行有效测试就显得越来越重要。其中内建自测试被认为是解决测试问题有效方法之一。文中提出了一种选择多个单元的重新播种 BIST 测试方法,实验结果表明该方法可以降低硬件开销。

**关键词:**内建自测试;重新播种;线性反馈移位寄存器;混合测试

**中图分类号:** TN407

**文献标识码:** A

**文章编号:** 1673-629X(2007)01-0233-02

## A Reseeding BIST Way by Selecting Multiple - Cell

MA Jun

(Educational Technology Dept., Anqing Teachers College, Anqing 246011, China)

**Abstract:** With the rapid development of IC technology, more and more cores are integrated into a chip. How to carry through the effective test looks more and more important. Built - In Self - Test is one of the most effective ways to solve test problem. A reseeded Built - In Self - Test way by selecting multiple - cell is proposed, which shows low hardware overhead by experiments.

**Key words:** built - in self - test; reseeded; linear feedback shift register; mixed - mode testing

### 0 引 言

随着芯片集成度的提高,对芯片的测试要求也越来越高,尤其是系统芯片(System-on-a-Chip, SoC)的出现,由于嵌入了各种芯核(core),使得测试数据急剧上升,导致测试费用和难度加大。从1980年B. Konemann等提出了内建自测试(Built-In Self-Test, BIST)<sup>[1]</sup>概念之后,关于BIST的研究就越来越受到人们的重视。BIST完全是以摆脱传统的ATE设备为目标的研究方法,主要的测试功能全部在芯片上实现。

在研究重新播种的内建自测试方面,R. Rajski等在1998年提出了再播种(reseeding)方案<sup>[2]</sup>,不过这个方案中一个种子只是对应了一个确定性测试矢量。如果被测电路中有许多抗随机性测试的故障,则需要的种子个数就很多。A. Yamani等于2003年在VLSI测试会议上提出了使用很少种子的重新播种测试方法<sup>[3]</sup>,给出了种子的排序算法,使得一个种子可以对应于几个确定性测试矢量,这样可以降低测试的硬件开销。同年,李立健等提出把难测故障的测试矢量作为种子<sup>[4]</sup>,并利用这些测试矢量中的不确定位进行存储压缩,此外对种子产生的测试序列进行最小化处理,使测试序列的长度较短,减少了测试矢量的施加时间。梁华国等在2004年提出了一种双重种子的压缩

技术<sup>[5]</sup>,采用编码折叠计数器种子作为LFSR种子,有效地利用折叠计数器和LFSR重新播种的优点,能显著地压缩确定性测试集的矢量数和矢量宽度。文中提出了一种选择多个单元的重新播种测试方法,该方法相对于文献[3]所提出的单个单元重新播种方法减少了硬件开销和测试序列长度。

### 1 算 法

文中所提出的选择多个单元的重新播种BIST测试方法主要是针对使用很少种子的重新播种BIST方法中只利用了线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)的一个单元作为输出向扫描链移入测试向量,没有充分利用到LFSR其他编码种子的能力这一不足进行的。虽说从LFSR任何一个单元输出测试矢量均可获得类似的矢量,但是当选用不同的触发器作为输出,产生某个测试矢量所需要的时钟个数不同,这就影响到测试序列的长短。一个较好的测试方法必须兼顾好测试硬件开销大小和测试序列长短的关系。

假设LFSR的级数为 $k$ ,扫描链的长度为 $n$ 。测试要从 $k$ 个LFSR单元中选择出 $p$ 个单元,在不同的测试阶段,这 $p$ 个单元分别作为输出向扫描链移入若干个长度为 $n$ 的测试矢量,这样可以充分利用到LFSR中多个单元编码种子的能力,使得一个种子可以对应更多的测试矢量。这里LFSR长度及连接多项式的选择与使用很少种子的重新播种内建自测试方法相同。测试时事先规定每个单元能输

收稿日期:2006-04-10

基金项目:安徽省教育厅自然科学基金项目(2006kj156c)

作者简介:马 俊(1962-),男,副教授,安徽安庆人,研究方向为VLSI。



出的最大测试矢量个数,各单元序号如图 1 所示。

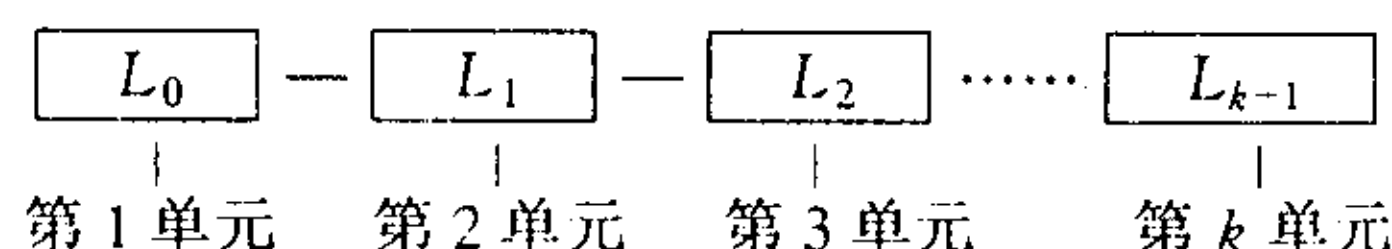


图 1 级数为  $k$  的 LFSR 单元序号图

测试开始时,先按照使用很少种子的重新播种内建自测试方法,选择 LFSR 的最后一个单元——第  $k$  单元作为输出,运行 LFSR 使它工作于随机测试状态,当运行了若干个时钟周期后,则可产生一定数量的测试矢量对电路进行检测,剩下一些难测故障。对难测故障,按照其确定性位数的多少由高到低排列,确定性位数越多的矢量越先为其计算种子。

设  $P$  为某被测电路的确定性测试矢量集合(元素按照其确定性位数的多少由高到低排列为  $p_1, p_2, p_3, \dots$ )。 $S$  为种子集合( $S$  初始时空集), $C$  为单元集合( $C$  初始时有一个元素  $k$ )。确定性测试步骤如下:

第一步:从集合  $P$  中选择确定位数最多的测试矢量  $p_1$ ,假设选取第  $i$  ( $1 \leq i \leq k$ ) 个 LFSR 单元作为输出向扫描链移入测试矢量,则根据 LFSR 的代数性质和求解种子方程的自由度为其计算种子。如果有多个单元都能满足要求,则哪个单元对应需要的时钟周期最短就选用哪个单元,并把这个单元的序号加入到集合  $C$  中,把对应的种子加入到集合  $S$  中,把  $p_1$  矢量从集合  $P$  中删除。

第二步:把第一步中计算的种子作为 LFSR 中的初始状态,然后使 LFSR 工作若干个时钟周期。在不超过事先规定的每个单元能输出的最大测试矢量范围内,看看是哪个单元的第几个矢量能与集合  $P$  中的  $p_j$  矢量兼容。哪个单元能产生  $p_j$  矢量,哪个单元就被选中。如果有多个单元的输出都能产生  $p_j$  矢量,就先选择集合  $C$  中已有的元素单元,如果之前的单元中没有该单元,则选择所需输出测试矢量个数最小的那个单元,并把该单元序号加入到集合  $C$  中,把确定性测试矢量集合  $P$  中的  $p_j$  删去,不用再为它计算种子了。没有找到兼容的话,就按照第一步中的方法为  $p_j$  矢量计算种子。

在前面两步的进行中,当遍及所有确定性测试矢量时得到的集合  $S$  即为最后的种子集合,得到的集合  $C$  即为最后的单元集合。

## 2 实验

实验仍采用混合测试方法。第一阶段使用随机性测试方法对电路进行测试,第二阶段对仍未能检测出的故障改用文中所提出的选择多个单元的重新播种测试方法。第一阶段对每个被测电路而言,借助于故障模拟实验,当发现电路的故障覆盖率不再提高时,记录输出的伪随机测试矢量的个数及未能检测的难测故障的个数等信息,结果如表 1 所示。

在为确定性测试矢量计算相应的种子前,先把测试矢量进行合并。结果如下:

(1) c432 合并后所需的矢量为:

- ① x0xxxxxxxx00xxxxxxxxxxxx00x11111x0
- ② 1xx11x11x011xxx11xx11xx11xx11x011101
- ③ xx111x111x1x1x1xxx11xx1x1x11xx101110
- ④ 1xxxx0x1xx1xx11xx0xxx0xxx0xxx0x100xx

(2) c499 合并后所需的矢量为:

- ① x0xx101x10x1101x1xx100x0xxxx101x11xx11x1x
- ② 1xxx0xxx1xxx11011xx11xxx101xx0xxx111xx1x
- ③ x1x1x0x011xxx0x11011101x10x1x01x111x110x1
- ④ xxx11x1110xx10111xx1000x1x111xx11x1x111x
- ⑤ 101xx011x011x011xx1100xx11110110x10011x11

表 1 随机测试的结果

电路名称	LFSR 的长度	随机向量数	剩余难测故障
c432	36	850	10
c499	41	1200	8

这里把如上提出的选择多个单元的重新播种测试方法用 C 语言程序实现,并对 ISCAS'85 基准电路中的 c432 和 c499 电路分别进行测试,检测这些电路的内部信号线上有无发生固定型故障。表 2 是实验结果。

表 2 选择多个单元的重新播种测试方法的实验结果

电路名称	LFSR	所需种子的个数	选择单元数	确定性测试产生的矢量数
c432	36	1	2	682
c499	41	2	4	8309

具体来说:

(1) 对基准电路 c432 进行研究,得到:

测试需要存储一个种子,选择两个单元。种子为: (10110101110111111011010110111011101), 这个种子载入 LFSR 后, LFSR 第 26 单元的第 238 个输出矢量与合并后的矢量①兼容,第 26 单元的第 240 个输出矢量与合并后的矢量④兼容,第 36 单元的第 1 个输出矢量与合并后的矢量②兼容,第 36 单元的第 341 个输出矢量与合并后的矢量③兼容。确定性测试需产生的矢量个数为  $341 \times 2 = 682$  个。

(2) 对基准电路 c499 进行研究,得到:

测试需要存储两个种子,选择四个单元。种子 1 为: (10011111101011001110111011100101100011111), 这个种子载入 LFSR 后, LFSR 第 26 单元的第 1570 个输出矢量与合并后的矢量④兼容,第 41 单元的第 1 个输出矢量与合并后的矢量③兼容;种子 2 为: (111110010010111110001100110111011010101), 这个种子载入 LFSR 后, LFSR 第 39 单元的第 1723 个输出矢量与合并后的矢量①兼容,第 5 单元的第 1467 个输出矢量与合并后的矢量②兼容,第 41 单元的第 1 个输出矢量与合并后的矢量⑤兼容。确定性测试需产生的矢量个数为  $1723 \times 3 + 1570 \times 2 = 8309$  个。

## 3 结束语

将文中所提出的选择多个单元的重新播种测试方法

(下转第 238 页)



### 3.2 同步策略

一旦一个工作项被列入了共享工作列表中,同步策略就决定了特定的工作流参与者如何访问这个工作项。

首先,可以使用层次的(hierarchical)或基于市场的(market-based)方法来调节待办工作项。如果是层次的,工作流作用服务可以把工作项直接分配给特定的参与者(这样就不需要共享工作列表了),也可以先把所有待办工作项分配给经理,然后再由经理手动地去把工作项分配给具体的参与者。前者在工作流管理系统可以优化日程表的情况下是很好的,但是后者可以减少分配的机械性,得到更多用户的赞同。如果实现的是基于市场的分配策略,工作流参与者就会去竞争获得执行某个活动的权利,可以通过拍卖或别的市场机制来完成。最后,还可以使用一个固定的日程表来完成工作项的分配。在大多数情况下,工作项被放在共享工作列表,参与者可以按“先到先服务”的规则来选择执行待办工作项。

工作项的分配可以完全由工作流管理系统自动地完成,工作流管理系统可以自动地选择参与者,赋予他执行活动的权利,并把工作项分配给他。在半自动的情况下,工作流管理系统可以建议哪些参与者可以有权利去执行活动,但是最终的分配还是由另一个决策实例,譬如小组的经理来完成。

待办活动的分配可以有推(Push)和拉(Pull)两种方式。在使用推的情况下,资源会向工作流系统发出信号表明他的存在。这在工作流作用服务可以很好地优化活动的进度表,并且活动本身都是一致的,大部分是没有用户参与的情况下是非常有效的。在拉的机制下,某个资源会在适当的时间请求下一个工作项,但该资源不必是可用的。如果某个资源已经被分配到一定数量的活动,那么推和拉两种机制的结合也是可以的,但是每个活动的执行顺序将由执行者来决定。

(上接第 234 页)

和文献[3]所提出的重新播种内建自测试方法做对比,分别使用两种方法对 ISCAS'85 基准电路中的 c432 和 c499 电路进行测试,得到表 3 所示的测试结果。

表 3 两种重新播种测试方法的比较

电路	LFSR 长度	所需种子个数		所需种子个数	
		文献[3]方法	文中方法	文献[3]方法	文中方法
c432	36	1	1	1231	682
C499	41	2	2	17961	8309

由表 3 可知,选择多个单元的重新播种测试方法在测试的硬件开销方面及测试序列的长度方面有一定优越性。对 c432 和 c499 电路进行检测,所需的种子个数与使用文献[3]方法相比,测试长度明显变短了,减少约 50%。研究发现,文中所提出的方法对部分电路还可减少种子个数,即降低测试的硬件开销。若对更大的电路进行测试,优越性会更明显。

### 4 总结和展望

资源管理策略是工作流系统的核心技术之一,它关系着工作流调度的效率和资源利用率。对资源有效的、灵活的建模将会越来越成为重要的主题,在分布式系统和协同工作中更是如此。文中就是以工作流生命周期为基础围绕着工作流应用中资源管理的问题来进行讨论的。在以人为本的流程中,工作流技术领域已经展现出了技术和组织机构方面的挑战。所以这方面已经引起了工作流供应商和用户的重视。作为供应商,可以做的工作包括提供广泛的对可用资源库的集成机制,维护资源信息的接口,支持不同的分配和同步策略。而作为用户,可以做的事情就是通过矫正资源模型和工作分配策略来改进工作流的应用,维护一个最合适于自己公司的组织结构和政策。

#### 参考文献:

- [1] Arkin A, Askary S, Jekeli F S, et al. Web Services Choreography Interface (WSCI) 1.0 [S]. Palo Alto, CA, USA: W3C, 2002.
- [2] Banerji A, Bartolini D, Chopell V, et al. Web Services Conversation Language (WSCL) 1.0 [S]. Palo Alto, CA, USA: World Wide Web Consortium (W3C), 2002.
- [3] Curbera F, Goland Y, Klein J, et al. Business Process Execution Language for Web Services, Version 1.0 [S]. BEA, IBM, Microsoft, 2002.
- [4] WfMC. Interface 1: Process Definition Interchange Process Model [M]. Winchester, UK: Workflow Management Coalition, 1999.
- [5] Wainer J, Kumar A, Barthelness P. Security management in the presence of delegation and revocation in workflow systems [M]. Campinas, Brazil: Universidade Estadual de Campinas, 2001.

#### 参考文献:

- [1] Konemann B, Mucha J, Zwichoff G. Built-In test for complex digital integrated circuits [J]. IEEE Trans on Computers, 1980, 15(3): 315-319.
- [2] Rajski R, Tyszer J, Zacharia N. Test data decompression for multiple scan designs with boundary scan [J]. IEEE Transactions on Computers, 1998, 33(47): 1188-1200.
- [3] Al-Ya A A, Smitra A, McCluskey E J. BIST rescoding with very few seeds [J]. IEEE Transactions on Computers, 2005, 40(13): 150-174.
- [4] 李立健, 赵瑞莲. 减少多种子内建自测试方法硬件开销的有效途径 [J]. 计算机辅助设计与图形学学报, 2003, 15(6): 662-666.
- [5] 梁华国, 蒋翠云. 使用双重种子压缩的混合模式自测试 [J]. 计算机研究与发展, 2004, 41(1): 214-220.